דוח מכין מעבדה 1

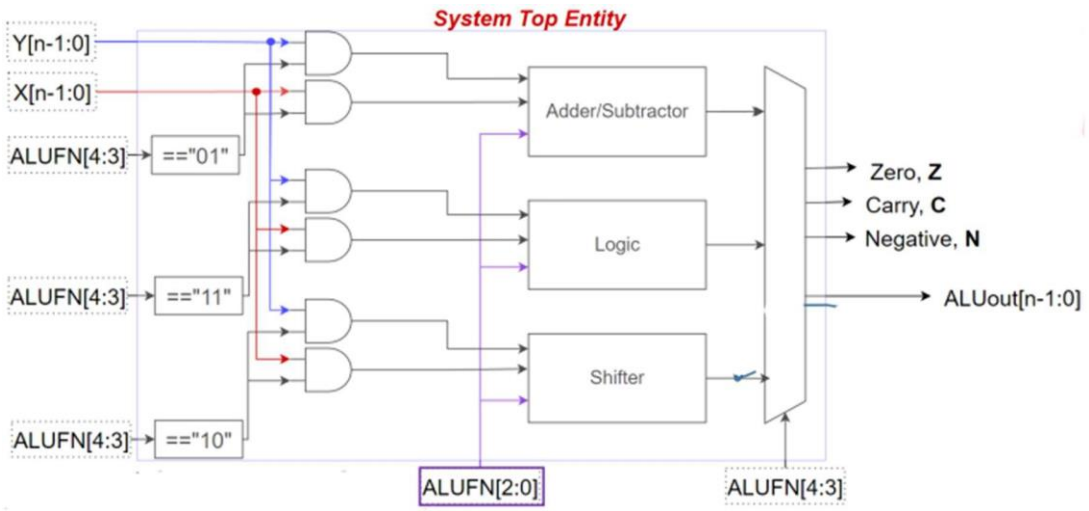
מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי חומרה

Roy Shani 315995258
Yanai Mason 206701591

מבוא

במעבדה זו נלמד יכולות בסיסיות בעולם החומרה המקבילית בשפת VHDL. במעבדה זו, נממש מערכת המכילה מספר רכיבים שונים, כאשר כל פעם מודול אחר מתבצע בנפרד לפי בחירת המשתמש.

המערכת שנממש נמצאת באיור הבא:



חלקי המערכת:

אות כניסה X.

אות כניסה Y.

קו בקרה ALUFN כאשר ביטים 3,4 כלומר 2 ביטי הMSB קובעים את הרכיב הנבחר באופן הבא:

- 01 הוא רכיב AdderSub.
- 10 הוא רכיב Shifter.
- 11 הוא רכיב Logic.

המודולים יכולים לבצע מספר פעולות שונות כתלות בשלושת הביטים הLSB בקו הבקרה (ביטים 0, 1, 2).

מוצאי המערכת:

- דגלי בקרה C (Carry), N (Negative), Z (Zero), V (Overflow)
- תוצאת פעולת הרכיב יוצאת בALUOUT

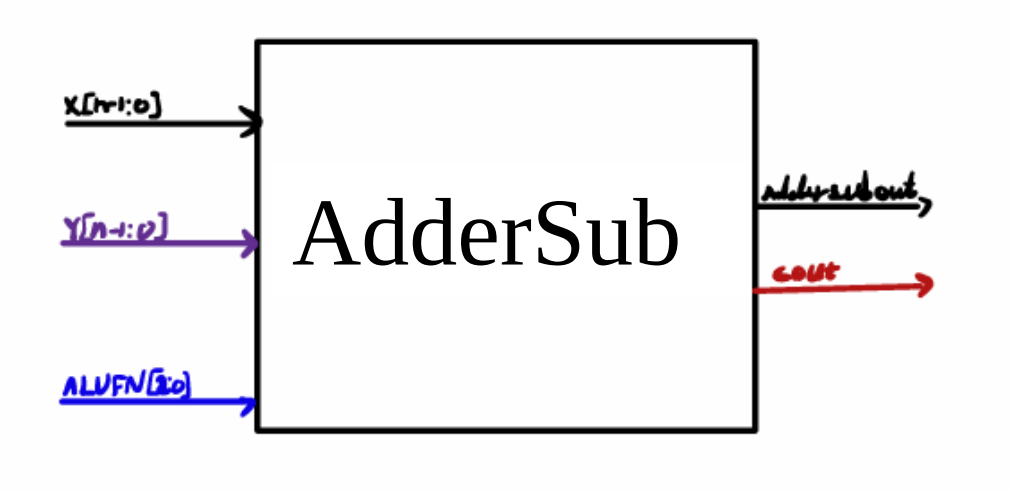
Function Kind	Decimal value	ALUFN	Operation	Note
Arithmetic	8	**01**000	Res=Y+X	
	9	**01**001	Res=Y-X	Used also for comparison operation
	10	**01**010	Res=neg(X)	
	11	**01**011	Res=Y+1	Increment of Y in one
	12	**01**100	Res=Y-1	Decrement of Y in one
Shift	16	**10**000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of q≜X(k-1 ..0) times Res=Y(n-1-q…0)#(q@0) ***When*** $k = log_2 n$
	17	**10**001	Res=SHR Y,X(k-1 to 0)	Shift Right Y of q≜X(k-1 ..0) times Res=(q@0)#Y(n-1...q) ***When*** $k = log_2 n$
Boolean	24	**11**000	Res=not(Y)	
	25	**11**001	Res=Y or X	
	26	**11**010	Res=Y and X	
	27	**11**011	Res=Y xor X	
	28	**11**100	Res=Y nor X	
	29	**11**101	Res=Y nand X	
	30	**11**111	Res=Y xnor X	

מודול AdderSub

תיאור

רכיב זה יופעל כאשר שני הביטים הMSB ב ALUFN יהיו "01". ויבצע את אחת מחמשת הפעולות הבאות בהתאם לכניסת 3 הביטים הLSB בALUFN באופן הבא:

- "000"- חיבור בין X ל Y
- "001"- חיסור בין Y ל X
- "010"- NEG(X)
- "011" – Inc Y
- "100" – Dec Y



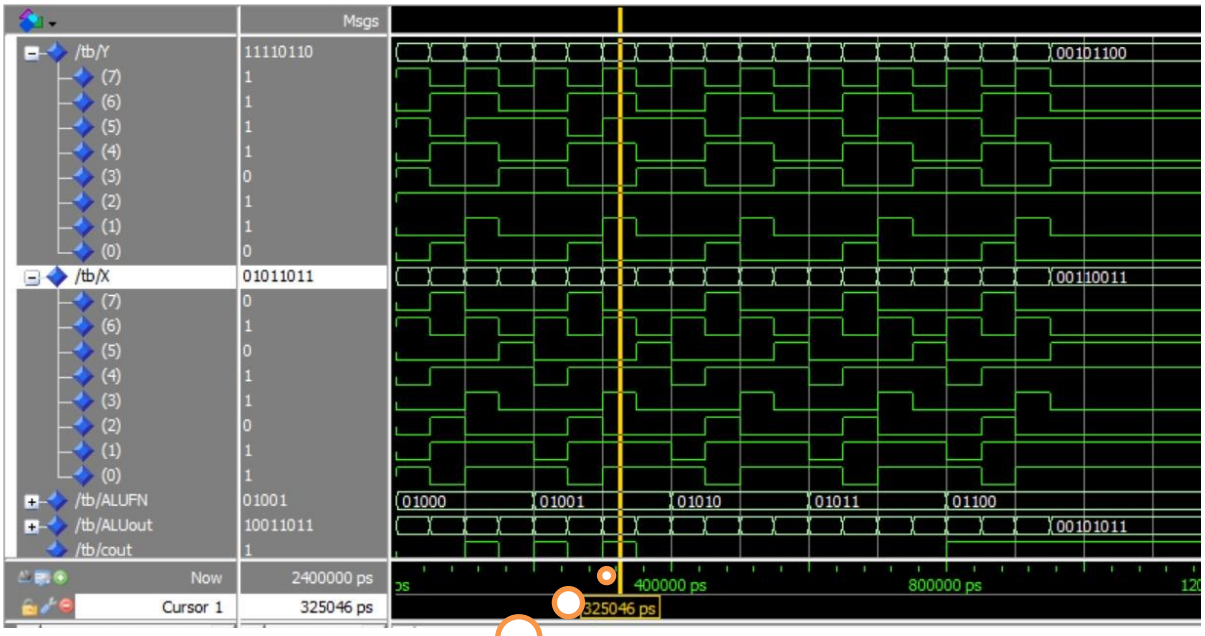
SUB_CONT אחראי על בחירת אחת מתת פעולות הרכיב.

מימוש בVHDL

1. הגדרנו את FA כרכיב
2. יצרנו סיגנלים עבור משתני עזר, למשל וקטור אפסים ואחדות
3. מייצרים את SUB_CONT על בסיס ביטי הבקרה
4. בוחרים את הוקטור X ע"פ ביטי הבקרה
5. בוחרים את הוקטור Y ע"פ ביטי הבקרה
6. סיגנל המכיל תוצאת XOR בין X לבין SUB_CONT
7. חיבור FA ראשון

8. חיבור יתר FA
9. הוצאת COUT

תוצאות הסימולציה



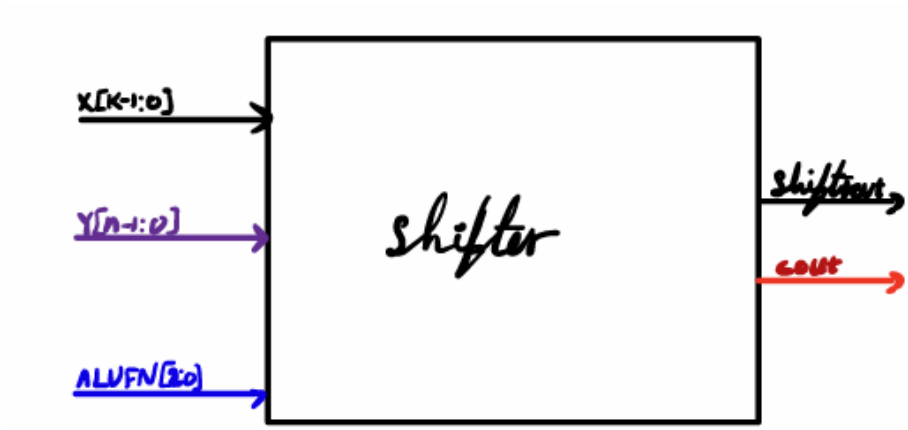
Y=-10 ,X=91,
ALU=01001(SUB), then
ALUout =10011011(-101)

מודול Shifter

תיאור

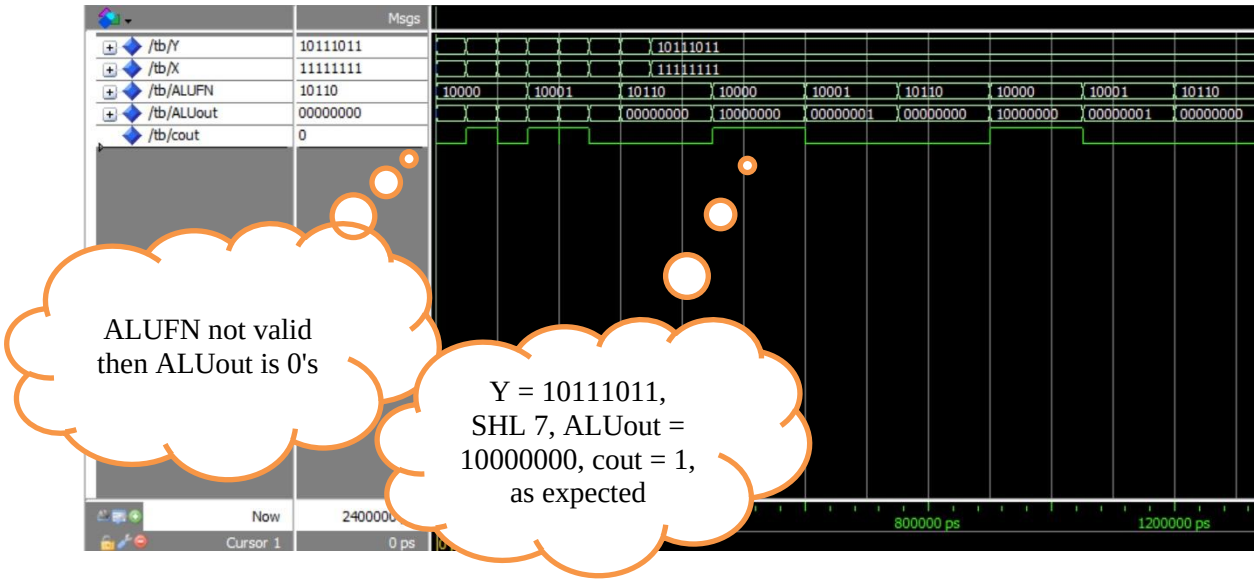
רכיב זה מופעל בהתאם לכניסת ALUFN ומבצע פעולת הזזה המבוססת על BARREL SHIFTER כאשר ביטים 0,1,2 מחליטים אם ההזזה תתבצע ימינה או שמאלה, "001", "000" בהתאמה. נפנה לרכיב זה כאשר שני הביטים הMSB של ALUFN יהיו "10".

הshifter מבצע הזזה של עד K רמות, כשבכל רמה מבצעים הזזה ימינה של Y לפי ערך 3 הביטים הLSB של X. במידה וברמה מסוימת לא מבצעים הזזה (כלומר ביט ה X-במיקום המתאים הוא 0), הרמה הבאה מקבלת את הווקטור כמו שהוא. במידה ומתבצעת הזזה, הווקטור מרופד באפסים משמאל לפי גודל ההזה .כדי לבצע הזזה שמאלה, אנחנו הופכים את הווקטור, מבצעים הזזה ימינה, והופכים אותו חזרה במוצא. לחישוב ה(bit carry)cout-אנחנו בודקים בכל רמה האם התבצעה הזזה, ואם כן שומרים את ביט ה cout - שיתקבל מהזזת הווקטור שנכנס לאותה רמה. במידה וברמה לא התבצעה הזזה, אז היא מקבלת את ה cout-של הרמה הקודמת. לבסוף מוציאים את ה cout-האחרון שקיבלנו, כי הוא מעודכן לפי ההזזה האחרונה שהתבצעה. המודול עובד לפי ערכי k,n גנריים ובעזרת פעולת generate על מנת לייצר חומרה כתלות בערכים אלו .במידה והוזן opcode שאינו מוגדר, הרכיב יוציא ווקטור אפסים כנדרש בהגדרת המטלה.

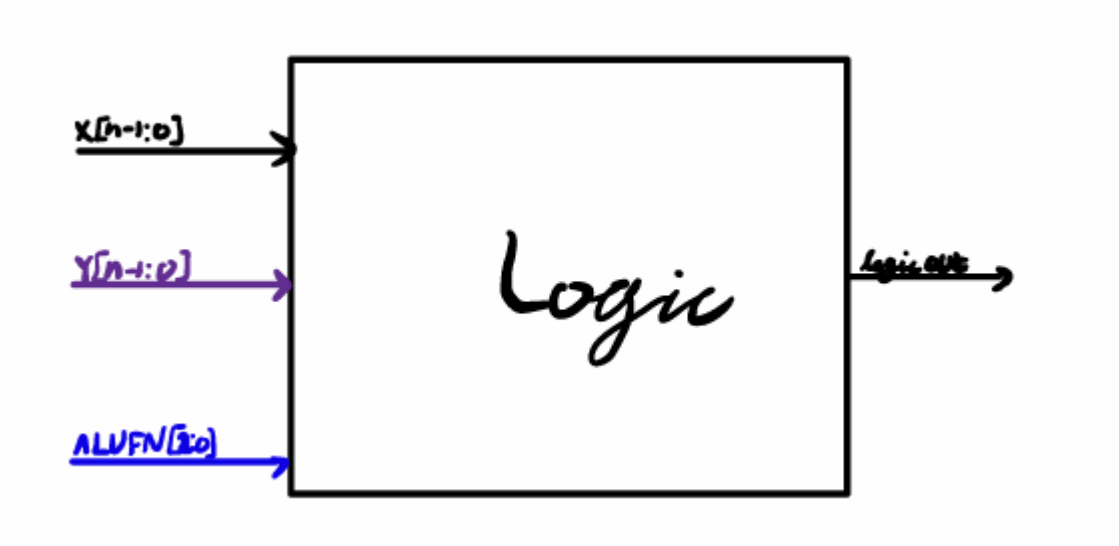


הערה: כאשר ההזזה מוגדרת ימינה נבצע את שלב 2 בצורה הפוכה ולבסוף לאחר סיום שלב 5 נהפוך את התוצאה הסופית על מנת לקבל הזזות כרצוי.

תוצאות הסימולציה



רכיב Logic



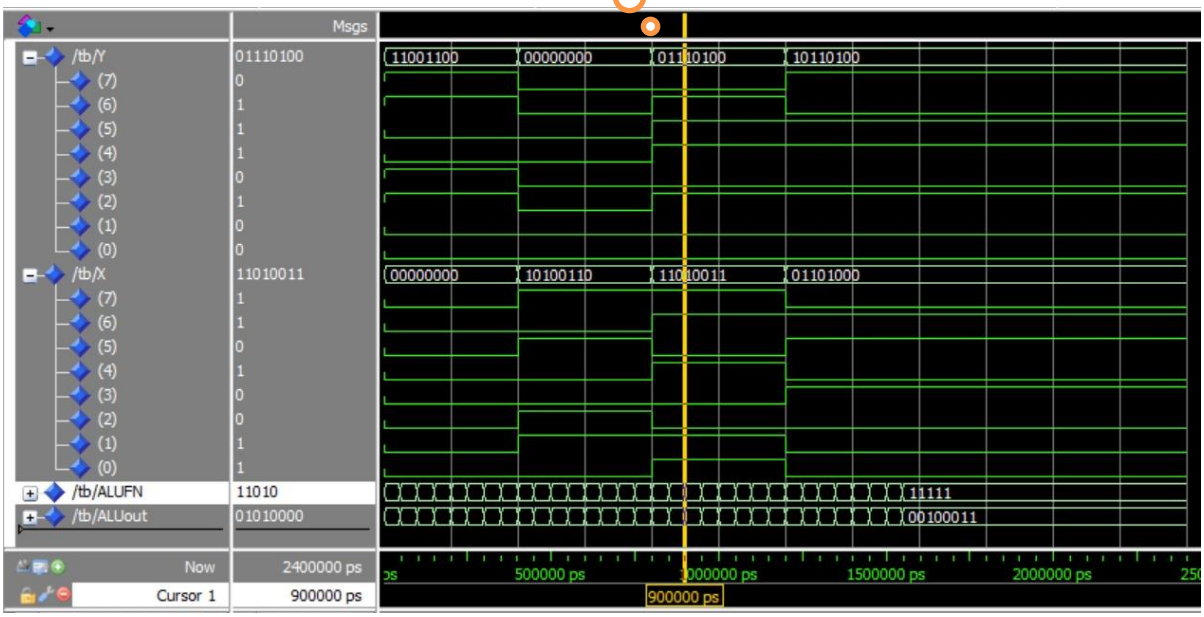
תיאור

ניכנס לרכיב זה כאשר שני הביטים הMSB של ה ALUFNיהיו "11".
רכיב זה מבצע פעולות לוגיות על וקטורים X ו Y ע"פ כניסת ALUFN:

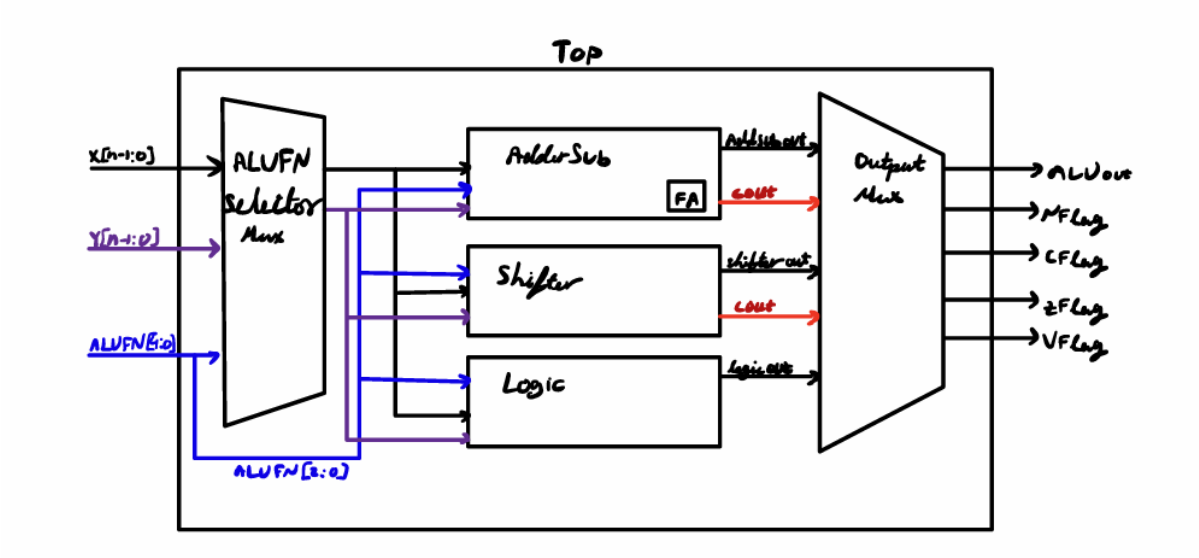
- NOT(Y) -"000"
- X OR Y -"001"
- Y AND X -"010"
- Y XOR X -"011"
- Y NOR X -"100"
- Y NAND X -"101"
- Y XNOR X -"111"

01110100 &
11010011 =
01010000 as
expected

תוצאות סימולציה



מודול Top



תיאור

רכיב ה -top הוא הרכיב העליון של המערכת, וממש בתוכו 2 פעולות מרכזיות:

1. עיבוד ראשוני של אותות הכניסה X,Y,ALUFN.

2 וקטורי הכניסה באורך n ביטים וALUFN באורך 6 ביטים כאשר 2 הביטים הmsb מגדירים את המודול שמבצע את הפעולה, ו3 הLSB מגדירים את הפעולה המבוצעת במודול.

מוצא: 3 זוגות של וקטורי Y,X, כאשר כל זוג מחווט אל תתי-המודולים AdderSub, Shifter, Logic. רק המודול דרכו תתבצע הפעולה המבוקשת (עפ"י כניסת ALUFN[4:3]) יקבל את Y,X הנכונים, 2 המודולים האחרים יקבלו וקטורי אפסים על מנת למנוע צריכת הספק מיותרת.

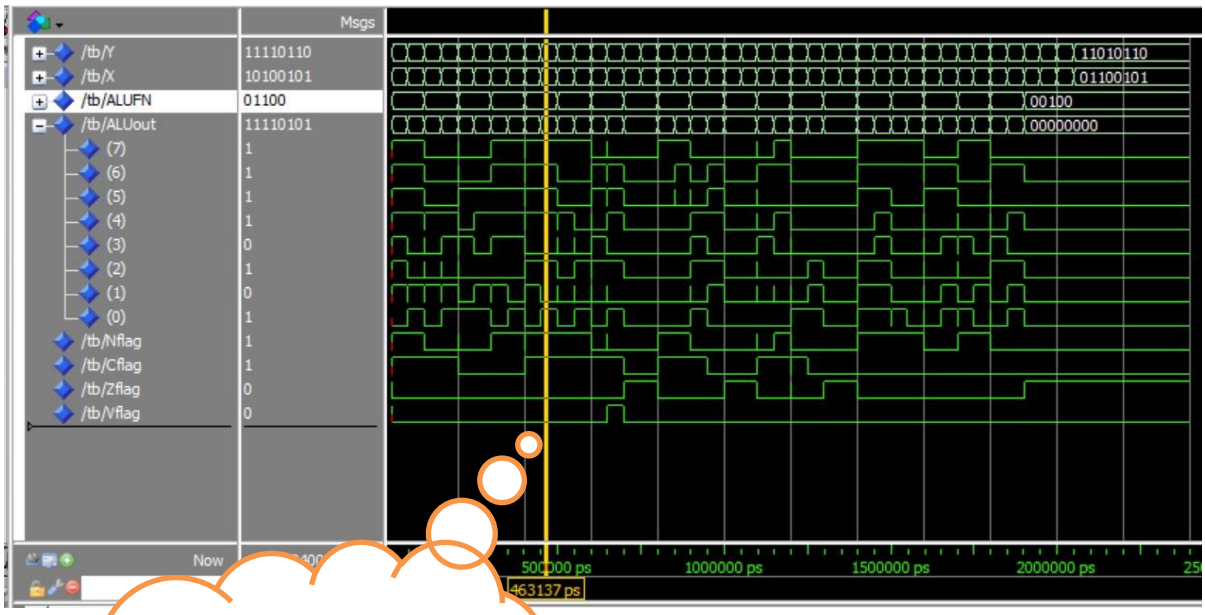
2. עיבוד סופי של אות המוצא ובדיקת דגלים

מבוא: 2 וקטור הכניסה המקוריים, ו3 וקטורי המוצא של תתי-המודולים, ביטי carry של תתי-המודולים Shifter & AdderSub, וקטור ALUFN[4:3].

מוצא: מבצע mux בין וקטורי המוצא של תתי-המודולים וביט ה-carry על בסיס ALUFN[4:3], מבצע פעולות לוגיות לבדיקת 3 הדגלים האחרים - עבור bit overflow בודק את הMSB של 2 וקטורי הכניסה המקוריים, MSB וקטור המוצא וה-ALUFN במטרה לזהות את ארבעת המצבים האפשריים לקבלת overflow.

עבור bit negative בודק את ה-MSB של וקטור המוצא. עבור bit zero משווה את וקטור המוצא עם וקטור אפסים.

תוצאות הסימולציה



For example ALUFN=01100 which mean AdderSub dec Y So Y = 11110110, ALUout = 11110101, as expected